2022 Boomi Deployment End User Info - For Public Distribution

FOR REFERENCE ONLY – No warrants made nor support offered.

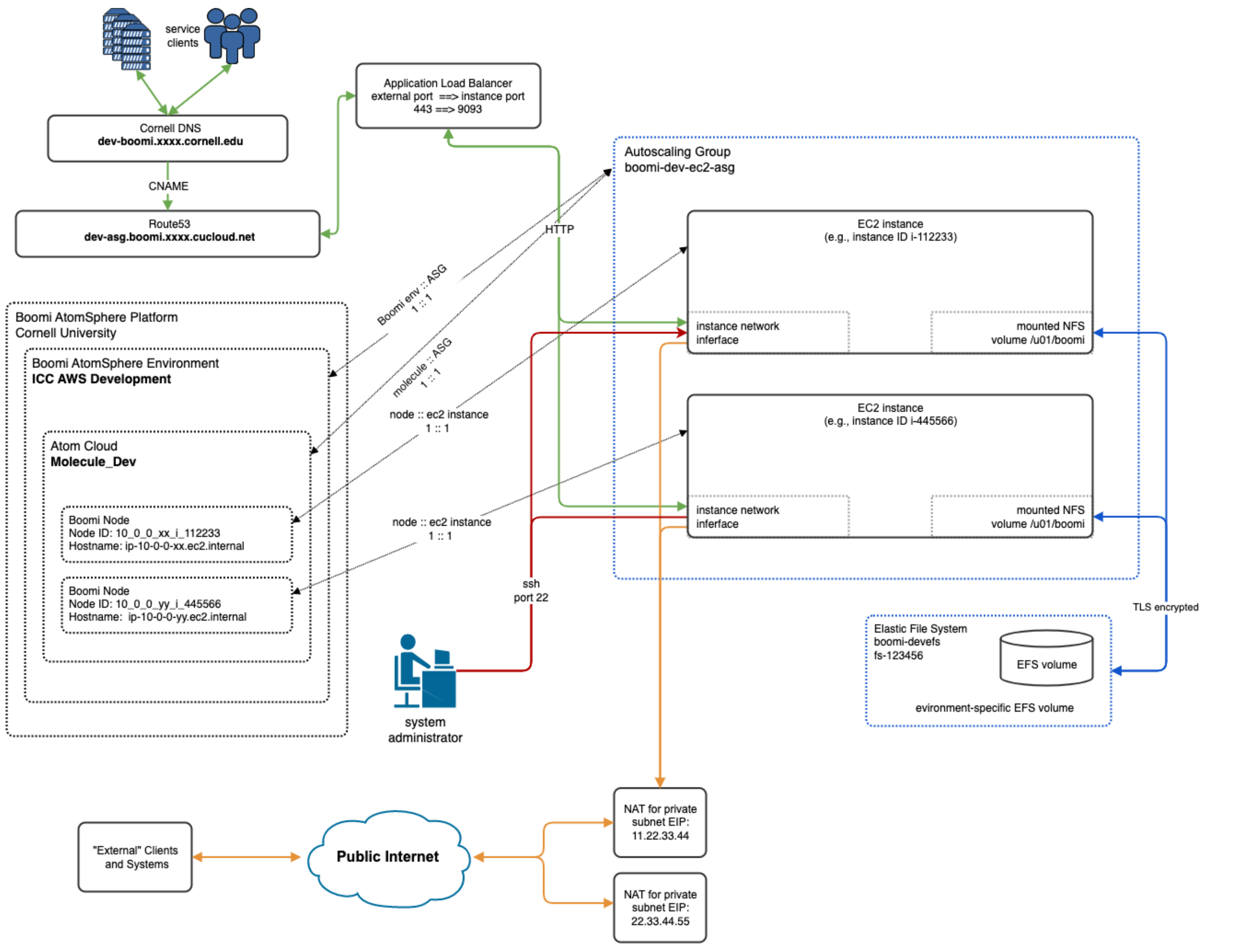
This documentation has been redacted/cleansed for public distribution for use in Cornell presentations about Boomi.

Support Escalation

When there are problems with the Boomi deployment that do not seem to be related to Boomi itself, please use the following contacts in order of escalation:

1. Administrators: xxx-admin@cornell.edu
2. Cloud Team
   a. Incidents (something is broken): xxx-incident@cornell.edu
   b. Requests: xxx-support@cornell.edu

Architecture


service clients
Application Load Balancer external port ==> instance port 443 ==> 9093
Cornell DNS dev-boomi.xxxx.cornell.edu
CNAME
Route53 dev-asg.boomi.xxxx.cucloud.net
Autoscaling Group boomi-dev-ec2-asg
HTTP
EC2 instance (e.g., instance ID i-112233)
instance network inferface
mounted NFS volume /u01/boomi
EC2 instance (e.g., instance ID i-445566)
instance network inferface
mounted NFS volume /u01/boomi
Boomi env :: ASG 1 :: 1
molecule :: ASG 1 :: 1
node :: ec2 instance 1 :: 1
node :: ec2 instance 1 :: 1
Boomi AtomSphere Platform Cornell University
Boomi AtomSphere Environment ICC AWS Development
Atom Cloud Molecule_Dev
Boomi Node Node ID: 10_0_0_xx_i_112233 Hostname: ip-10-0-0-xx.ec2.internal
Boomi Node Node ID: 10_0_0_yy_i_445566 Hostname: ip-10-0-0-yy.ec2.internal
ssh port 22
system administrator
TLS encrypted
Elastic File System boomi-devefs fs-123456
EFS volume
evironment-specific EFS volume
NAT for private subnet EIP: 11.22.33.44
NAT for private subnet EIP: 22.33.44.55
"External" Clients and Systems
Public Internet

draw.io source: Boomi-ASG-Architecture.for-distribution.drawio

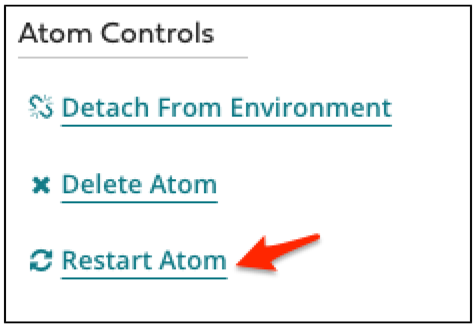
Factoids

Boomi Deployments (Environments)

Deployment	AWS Account	Environment Tag	VPC	VPC CIDR	cucloud.net Endpoint	cornell.edu Endpoint(s)	Endpoint Accessibility
dev	123456789012	dev	vpc-111111 xxxxxx-test-vpc	10.0.0.0/22	https://dev-asg.boomi.xxxx.cucloud.net	n/a	Cornell private network
test	123456789012	test	vpc-222222 xxxxxx-prod-vpc	10.0.4.0/22	https://test-asg.boomi.xxxx.cucloud.net	https://dev-boomi.xxxx.cornell.edu https://test-boomi.xxxx.cornell.edu	public internet
prod	123456789012	prod	vpc-222222 xxxxxx-prod-vpc	10.0.4.0/22	https://prod-asg.boomi.xxxx.cucloud.net	https://boomi.xxxx.cornell.edu	public internet
sandbox	234567890123	sandbox	vpc-333333 xxxxxx-sandbox-vpc	10.0.8.0/22	https://sandbox-asg.boomi.xxxx.cucloud.net	n/a	Cornell private network

Atom Restart

In the ASG-based Boomi environment, when the **Restart Atom** button is clicked in the Boomi web UI, each node restarts in a rolling fashion, similar to in the OpsWorks-based deployment.



The ASG-based environment does not have the same problem restarting nodes that occurred with the container-based nodes.

Additional detailed information about atom **stop**, **start**, and **restart** behavior is documented in Boomi Actions and Behaviors.

Boomi Node ID

Boomi **Node ID** is derived from the AWS EC2 instance ID and IP address of the instance. The format of the Node ID is **IPADDRESS_INSTANCEID**, with "." and "-" replaced with "_".

E.g. The Node ID for the boomi node with IP address **10.0.0.xx** running on EC2 instance **i-062cdcdac0e5d4f29** is **10_0_0_xx_i_062cdcdac0e5d4f29**.

Boomi Host Name

Boomi **Host Name** is related to the private IPv4 address assigned EC2 instance for the node. E.g., Boomi node with Host Name ip-10.0.0.xx.ec2.internal has IP address 10.0.0.xx.

EC2 Instance Name

The **Name** of the EC2 instance reported in the EC2 console has form **boomi-ENV-ec2** where ENV is one of **dev**, **test**, **prod**.

All EC2 instances for a given Boomi environment will be named the same.

IPv4 Addresses

The data in this table reflect only the **test** and **prod** environments! The **dev** and **sandbox** environments use other VPCs and thus other CIDR blocks.

Resource Type	Description	CIDRs	Notes
EC2 Instance	IPv4 CIDR of EC2 instances	10.0.0.0/26	
NAT Gateway	IPv4 CIDR of NAT Gateway used by Boomi EC2 instances	11.22.33.44/32 22.33.44.55/32	These public IPv4 addresses will be used when Boomi-nodes reach out to the public internet.
Load Balancers	The Load Balancers acting as the public front door to the Boomi web API interfaces.	0.0.0.0/0	The public IPv4 addresses used by the Load Balancers are not static. They could be assigned any IP public address owned by AWS, and will change over time. If Boomi clients need to whitelist these Load Balancers, they would have to do so using the domain name of the Load Balancers. I.e., **test-boomi.xxxx.cornell.edu**, **boomi.xxxx.cornell.edu**.

Timeline for Launching New Nodes

Here is the rough timeline you can expect when new EC2 instances are launched by the ASG.

Total Elapsed Time **minutes:seconds**	**What**
0:00	CloudFormation stack update is submitted to increase the **DesiredEC2InstanceCountParam** parameter value.
0:30	ASG configuration is updated and the ASG launches a new EC2 instance
5:30	• The new EC2 instance is running and configured (all packages installed). • Atom service on instance is started.
5:40	Boomi node on new EC2 instance is fulling running.

Boomi EC2 Host Directory Structure

/home /boomi /local	working data local storage	This directory is local to the instance and is not accessible external to the instance.
/home /boomi	Boomi user directory	This directory is local to the instance is not accessible external to the instance.
/u01 /boomi/	root of shared molecule directory structure	This directory is mounted from a shared Elastic File System. At minimum all the instances that makeup the same Molecule will share this directory. However, other Molecules may as well, depending on how they are deployed.
/vat/tmp	Boomi/Java temporary directory	This is the Java temp directory, **java.io.tmpdir**. Configured in **bin/atom.vmoptions**. This directory is local to the instance and is not accessible external to the instance.

Cron Jobs on Boomi EC2 Instances

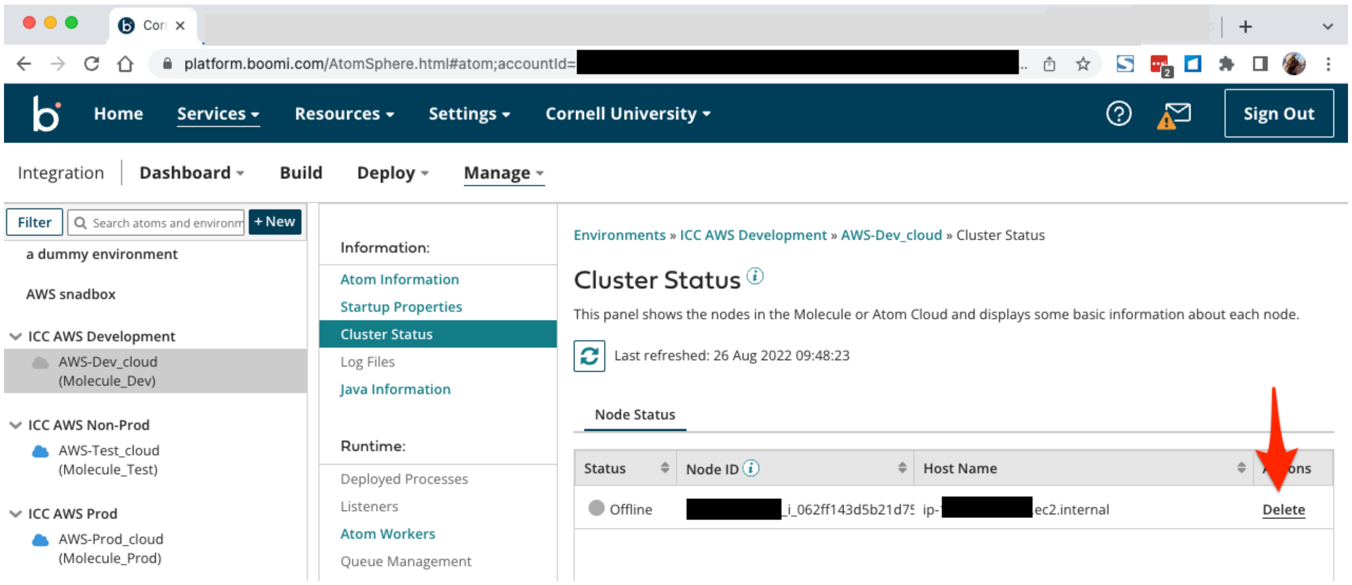
These cron jobs are configured as part of the instance launch cloud-init process.

- **/etc/cron.d/clean_local_space**
  - runs daily at 1am
  - Calls /u01/boomi/bin/clean_local_space.sh, which is managed by Administrators
- **/etc/cron.hourly/boomi-initial-hosts**
  - runs hourly
  - Updates the instance specific container properties file: **/u01/boomi/Cloud_Molecule_XXX/conf/container.IPADDRESS_INSTANCEID.properties** file.

Procedures (for Boomi Integrations Team)

Cleaning Up Cluster Status

Whenever an instance running a molecule node is stopped or deleted, the Cluster Status page on GUI on the Boomi Platform will show a line for the defunct node with gray status (see screenshot below). It is always safe to use the pulldown action menu under the "gear" to "Remove from Table", thus cleaning up the cluster list.


platform.boomi.com/AtomSphere.html#atom;accountId=[redacted]

a dummy environment

AWS snadbox

ICC AWS Development
- AWS-Dev_cloud (Molecule_Dev)

ICC AWS Non-Prod
- AWS-Test_cloud (Molecule_Test)

ICC AWS Prod
- AWS-Prod_cloud (Molecule_Prod)

Information:
- Atom Information
- Startup Properties
- Cluster Status
- Log Files
- Java Information

Runtime:
- Deployed Processes
- Listeners
- Atom Workers
- Queue Management

Environments » ICC AWS Development » AWS-Dev_cloud » Cluster Status

Cluster Status

This panel shows the nodes in the Molecule or Atom Cloud and displays some basic information about each node.

Last refreshed: 26 Aug 2022 09:48:23

Node Status

Status	Node ID	Host Name	Actions
Offline	[redacted]_i_062ff143d5b21d75	ip-[redacted]ec2.internal	Delete

Escalation of Methods for Restarting a Boomi Molecule

There are a number of ways to restart a molecule – either a specific molecule node, or all the nodes in a molecule.

Level of escalation	What action is taken	What happens?	How to do it	Who can do it
1 - minor /easy	restart all the atom processes on all molecule nodes	the atom service (Java process) on all the EC2 instances forming a molecule is stopped then started	Use Boomi Platform: "Atom Information" "Restart Atom"	Boomi team members with access to the Boomi platform GUI
2	restart the atom service running on a particular node	the atom service (Java process) on the target EC2 instance is stopped then started	Login to the EC2 instance and **sudo systemctl restart atom**	Administrators, Cloud Team
3	reboot a particular EC2 instance	the atom service (Java process) is gracefully stopped (if possible), the instance is rebooted, the atom service is started	Use the AWS EC2 Console to find the target instance and reboot it.	Administrators, Cloud Team
4 - major	deploy a replacement EC2 instance; terminate the problem instance EC2 instance	the new EC2 instance joins the molecule as a new node; the problem instance leaves the molecule cluster	See procedures in **Managing Auto Scaling Groups** section below.	Administrators, Cloud Team

Connecting to Boomi Shared Molecule Files with SFTP

Boomi Molecules in AWS use a shared AWS Elastic File System (EFS), one EFS for production, and one EFS for dev/test. Integration team members can access these using SFTP, either directly from the command line or with the aid of a SSH/SFTP client.

SFTP Client Configuration

See SFTP and Cloud Object Store Clients - OS X and Windows for a list of SFTP clients compatible with this SFTP system. Some of these clients allow systems to be integrated into native Windows or OS X file explorer interfaces giving a user experience similar to mounting a remote drive.

SFTP endpoint:	sftp.xxxx.cucloud.net (port 22)
Allowed users:	Any user that is part of the **CIT-boomi-adm** AD group. This AD group is managed by the DBA/Integrations Team
Credentials:	standard Cornell netid and password (Two-Step Login required)

Boomi SFTP Directory Structure

SFTP EFS Directory Structure	Test Molecule Host EFS Directory Structure	Prod Molecule Host EFS Directory Structure
/cu-odaa/molecule_test	/u01/boomi	n/a

/cu-odaa/molecule_test/security	/u01/boomi/security	n/a
/cu-odaa/molecule_test/Cloud_Molecule_Test	/var/boomi/Cloud_Molecule_Test	n/a
/cu-odaa/molecule_prod	n/a	/u01/boomi
/cu-odaa/molecule_prod/security	n/a	/u01/boomi/security
/cu-odaa/molecule_prod/Cloud_Molecule_Prod	n/a	/var/boomi/Cloud_Molecule_Prod

User and Group Ownership and Privileges

Containers running Boomi Molecules use the "boomi" user (uid = 1000) and the "boomi" group (gid = 1000). Molecule processes are configured to use **uma sk 002 for** files and directories they create. As such, default owner:group is boomi:boomi and the group will have the same privileges to as the boomi user.

SFTP users accessing files and directories have their user id assigned from Cornell AD, but are forced into group with id 1000 as their primary group when they connect to an SFTP system. This configuration, in addition to the umask and group id used by Boomi containers, should allow fairly SFTP users unhindered access to read and write files in the filesystem used by Boomi molecules. In addition, SFTP users have umask set to 002 as well, so that any group-level permissions set on files they upload should remain intact. Note that this configuration does NOT set read and write privileges for the group on a file uploaded if the file had only read privileges for the group on the source system. Some SFTP clients have features that allow privileges to be applied to files uploaded from the client system. Files uploaded to the EFS via SFTP will have group id set to 1000 since that is the primary group for SFTP users.

Metrics, Alarms, Dashboards, and Logging

This section describes features available to the Boomi Team in the AWS Console. The team will login to the AWS Console using the **shib-integrations** role. (These same features are available to AWS Console users with the **shib-admin** role.)

Via **shib-integrations**, the Boomi team will have read-only access to the following features:

- Metrics from Boomi EC2 instances, Elastic File System (EFS) volumes, and Elastic Block Storage (EBS) volumes
  - Dashboards showing these metrics
  - Ad-hoc charting for these metrics
- Raw log file contents, both generated by Boomi and by the EC2 hosts
- Log query capabilities using CloudWatch Logs Insights
  - The one set of "write" permissions in the Boomi-related privliges in the **shib-integrations** role is the ability to save and manage queries.
- Visibility of Alarms, including status, history, and configuration.

AWS Console Login

The direct URL to login is https://signin.aws.cucloud.net/.

Metrics

Dashboards

- List of all dashboards
  - The Boomi team can drill into and view any dashboard with a name prefixed by **"boomi-"**
  - production
    - EC2, EBS, Load Balancer info: boomi-prod-ec2-dash
    - EFS info: boomi-prod-efs-volume-fs-123456
  - test
    - EC2, EBS, Load Balancer info: boomi-test-ec2-dash
    - EFS info: boomi-test-efs-volume-fs-234567

Alarms

- List of all alarms
  - production alarms
  - test alarms

Log Access

Log Destinations

Several key logs from ASG EC2 instances are captured to CloudWatch logs. They can be found in separate Streams within CloudWatch Log Groups named **/boomi/asg/ENV**.

CloudWatch Log Groups:

- /boomi/asg/dev
- /boomi/asg/test
- /boomi/asg/prod

Instance File	CloudWatch Log Stream
/u01/boomi/Cloud_MOLECULE/logs/*.container.ATOM_LOCALHOSTID.log	/INSTANCE_ID/IP_ADDRESS/boomi-container.log
/u01/boomi/Cloud_MOLECULE/logs/*.shared_http_server.ATOM_LOCALHOSTID.log	/INSTANCE_ID/IP_ADDRESS/boomi-shared-http-server.log
/var/log/syslog	/INSTANCE_ID/IP_ADDRESS/var/log/syslog
/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log	/INSTANCE_ID/IP_ADDRESS/cloudwatch-agent.log
/var/log/cloud-init-output.log	/INSTANCE_ID/IP_ADDRESS/var/log/cloud-init-output.log
/var/log/cloud-init.log	/INSTANCE_ID/IP_ADDRESS/var/log/cloud-init.log

Searching Logs using CloudWatch Logs Insights

```
filter @logStream like '/i-0dc6faa3f8e3afaf5/10.0.0.186/'
| filter @message like /(?i)Exception/ or @message like /(?i)Warn/ or @message like /(?i)Error/ or @message
like /(?i)Severe/
| fields @timestamp, @message
| sort @timestamp desc
```

- Use the **@LogStream like** filter to focus on all the logs from a specific instance (e.g., **/i-0dc6faa3f8e3afaf5/10.0.0.186/**), or from a specific type of log (e.g, **boomi-container.log**).
- The like **/(?i)Exception/ filter** will match "Exception" or "exception".

See complete CloudWatch Logs Insights query syntax in AWS documentation.

Shell Access to EC2 Instances (For Engineers)

SSH

- SSH is enabled for the EC2 instances, running on the standard port (22).
- Security Groups are configured to allow SSH traffic from a restricted CIDR block (xx.xx.xx.xx/xx).

To connect to an instance using SSH:

```
ssh -i ~/.ssh/oracle.pem boomi@xx.xx.xx.xx
```

SSM Session Manager Command Line Access

All Boomi EC2 instances have Session Manager enabled. This feature uses the SSM agent running on the instances to provide shell access to the system. The user experience is similar to SSH access, but can be initiated from the AWS web console.

- Starting a session from the EC2 console
- Starting a session using the AWS CLI

Session Manager sessions log you into the instance using the linux user named **ssm-user**. That user has **sudo** privileges so you can perform command using the **root** or **boomi** linux user.

Atom Service Management (For Engineers)

The main Boomi process on EC2 instances is the **atom** linux service.

If you have an SSH/SSM terminal session into a Boomi EC2 instance you can use the following commands to manage the **atom** service.

- **sudo systemctl restart atom**
- **sudo systemctl status atom**
- **sudo systemctl start atom**
- **sudo systemctl stop atom**
- **sudo journalctl -u atom** see /var/log/syslog entries relevant to the atom service

Resource Management Procedures (For Engineers)

The following phrases are used interchangeably in the text below:

- node Boomi node EC2 instance
- auto scaling group ASG

⚠ The names for the CloudFormation stacks involved below still include **ecs** in the name even though, eventually, ECS will not be involved. It would be unnecessarily disruptive to destroy the old CloudFormation stacks and create new ones with more sensical names.

How To Change...

Setting to be Changed	What Controls This	Process to Follow
Root EBS Volume Size	**RootEBSVolumeSizeOverrideParam** parameter	(No template changes required.) 1. Update the relevant CloudFormation Stack to set the new parameter value. 2. Once the Stack is updated, any new EC2 instances launched by the ASG will use the new volume size.
EC2 Instance Type	**InstanceTypeParam** mapping	1. Edit the **60-ecs.yaml** CloudFormation template to set the new Instance Type for the relevant environment (i.e. dev, test, prod) 2. Update the CloudFormation Stack with the new template; no stack parameter changes are required. 3. Once the Stack is updated, any new EC2 instances launched by the ASG will use the new instance type. 4. Commit the new **60-ecs.yaml** back to the Git repo https://github.com/CU-CommunityApps/xxxx
Number of Instances Running	**DesiredEC2InstanceCountParam** parameter	See the **Managing Auto Scaling Groups** section below.

Managing Auto Scaling Groups

- The number of EC2 instances running is determined by the configuration of the relevant **boomi-ENV-ec2-asg** Auto Scaling Group. But, be aware that CloudFormation stack parameters controls the ASG.
- There is a one-to-one correspondence between Boomi nodes and EC2 instances in the ASG.

ASG Scale-In Protection

When managing EC2 instances that are part of an ASG, **Scale-In Protection** is sometimes useful to use. Scale-in protection is a per-instance setting that can be used to prevent an ASG from terminating an specific instance that you wish to remain untouched. In the Boomi ASG deployment, scale-in protection is initially **off** (or disabled) for EC2 instances.

A use case where scale-in protection may be useful is when reducing the total size of the ASG. If you wish to make sure that the Boomi cluster leader or a node that is running a long process isn't the instance chosen by the ASG to be terminate, you can set scale-in protection on the node(s) to be preserved. The ASG will choose other node(s) to terminate to reach the target ASG size.

⚠ If you set scale-in protection to protect instances from termination, be sure to remove the scale-in protect after the operation (i.e., ASG size reduction) is complete.

Increase the Total Number of EC2 Instances

This is controlled by the **DesiredEC2InstanceCountParam** parameter in the relevant **boomi-ENV-ecs** CloudFormation stack.

1. Navigate to the relevant **boomi-ENV-ecs** CloudFormation stack:
   - boomi-dev-ecs stack
   - boomi-test-ecs stack
   - boomi-prod-ecs stack
2. Perform a stack update, increasing the **DesiredInstanceCountParam** parameter to the desired target, up to a hard-coded limit of 10 instances.
3. Complete the CloudFormation Stack update dialogs and wait for the stack update to complete. The stack update will change the configuration of the ASG.
4. After CloudFormation indicates that the stack has been updated, you will need to wait (briefly) for the ASG to recognize the need for additional instance(s) and launch more instance(s).

The newly launched EC2 instance will automatically join the relevant Boomi environment/cluster.

Reduce the Total Number of EC2 Instances

If you wish to prevent the ASG from terminating particular instances, you can set **Scale-in Protection** on those instances prior to changing the ASG size. But, be sure that enough instances have **Scale-in Protection** disabled so that the ASG can reach the new target ASG size. E.g., if the current ASG size is 3 and the target ASG size is 1, then at most one instance can have Scale-in Protection set; otherwise, the ASG has no way to reach the target size.
If you set Scale-in Protection for any instances, it is wise to remove Scale-in Protection once the resizing process is complete.

1. Navigate to the relevant **boomi-ENV-ecs** CloudFormation stack:
   - boomi-dev-ecs stack
   - boomi-test-ecs stack
   - boomi-prod-ecs stack
2. Perform a stack update, reducing the **DesiredEC2InstanceCountParam** parameter to the desired number.
3. Complete the CloudFormation Stack update dialogs and wait for the stack update to complete. The stack update will change the configuration of the ASG.
4. After CloudFormation indicates that the stack has been updated, you will need to wait (briefly) for the ASG to recognize the need to remove instance(s).

If the ASG terminates the Boomi node that is currently the head/leader of the cluster, the remaining nodes will determine a new head/leader

Automatic EC2 Management

ASGs can be configured to scale up or down automatically, based on metrics or based on time. E.g., the **dev** environment will have (an) instance(s) running on weekdays during normal work hours.

Additional automatic ASG scaling can be configured by the Cloud Infrastructure & Solutions Engineering Team (i.e., the CIT Cloud Team).

Replace a Single, Specific EC2 Instance

This process necessarily also retires the Boomi node running on the target EC2 instance.

1. Navigate to the instance inventory in the EC2 console.
2. Check the box in row containing the instance you wish to replace.
3. Select **Instance state Terminate instance**
4. In the **Terminate instance?** dialog, review the instance ID and name of the target instance. If correct, click on **Terminate**.
5. Now, wait. Within seconds/minutes the ASG will recognize that the targeted instance is gone, and will start a new instance to replace it.

The newly launched EC2 instance will automatically join the relevant Boomi environment/cluster.

Refresh (Replace) All EC2 Instances

This process necessarily also retires the Boomi node running on all the replaced EC2 instances.

1. Navigate to the relevant ASG.
   a. boomi-dev-ec2-asg
   b. boomi-test-ec2-asg
   c. boomi-prod-ec2-asg
2. Verify that you are looking at the correct ASG.
3. Click on the **Instance refresh** tab.
4. Click on the **Start Instance refresh** button.
5. Uncheck the **Enable skip matching** checkbox.
6. Click on **Start instance refresh**.

Using the default setting of 90% for **Minimum healthy percentage** and 30 seconds for **Instance warmup** leads to the ASG first terminating one (random) instance at a time and restarting its replacement before terminating the next instance. In once live test, we saw ~3 minutes in between the launching of new EC2 instances.

Restoring EFS Backups

The **prod** and **test** EFS volumes are manually configured to be backed up twice daily.

There are two options for the scope of files to be restored from a backup:

- restore the entire file system
- restore a specific file or directory

There are also two options for the location of the restoration:

- restore the target files to a new file system
- restore the target files to the original filesystem, under a new root-level directory.
  - There are no options to restore files to their original location!

See AWS documentation for Restoring an Amazon EFS file system.

If at all possible, restore files to the original file system and copy/move files as needed.

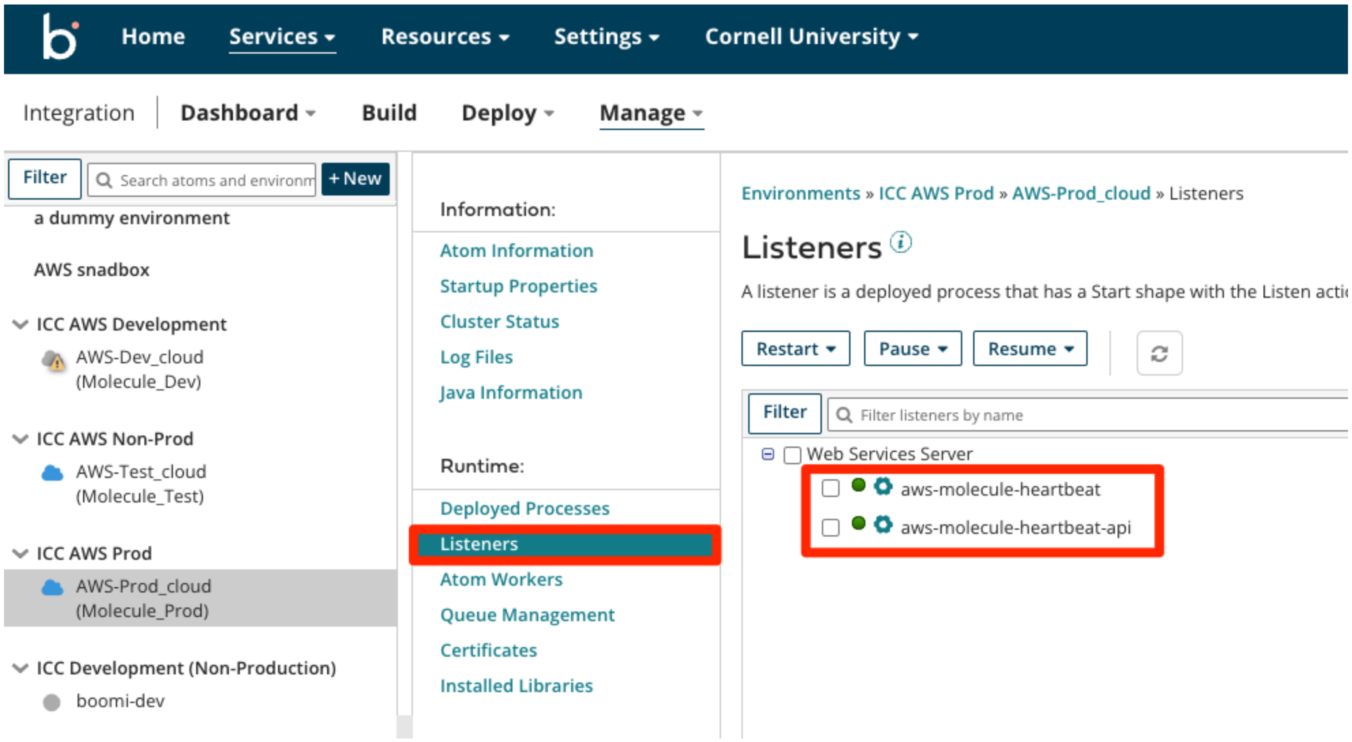
⚠️ If you decide to restore to a new EFS file system, there are several places in the CloudFormation-based deployment that will need to be updated with the ID of the new file system. At a minimum, restoring to a new EFS file system will require that new EC2 instances be launched for the environment where that has happened. In this situation, you probably will need to engage the CIT Cloud Team for assistance.

Testing Shared Web Service Web Service Configuration

A simple custom Boomi heartbeat web service that can be deployed to specific environments in order to test proper load balancer, certificate, and port configurations. That heartbeat web service uses a simple process (aws-molecule-heartbeat) and simple API (aws-molecule-heartbeat-api). This process and API is little more than a REST version of "Hello World".

Deploy this API and process as needed for testing. It is **not** required to be deployed to an environment for normal functioning of the AWS ASG deployment.

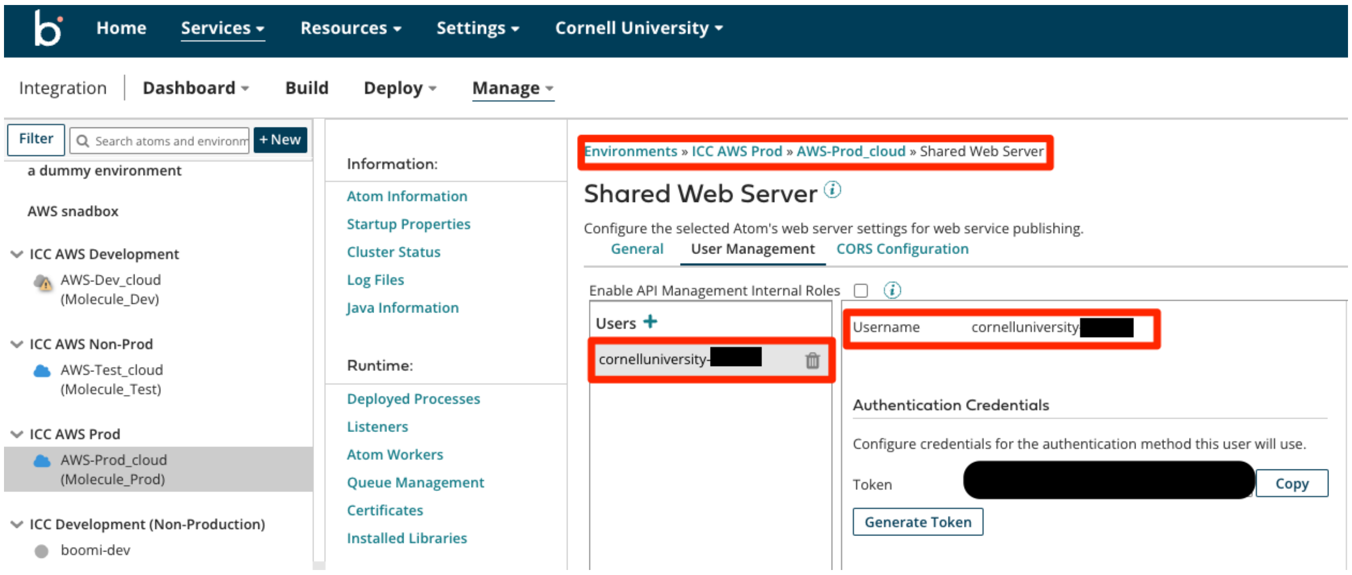
Deployed Listeners



A listener is a deployed process that has a Start shape with the Listen acti

Web Services Server
- aws-molecule-heartbeat
- aws-molecule-heartbeat-api

REST User Configuration



Configure the selected Atom's web server settings for web service publishing.

Enable API Management Internal Roles ☐

Users: cornelluniversity-[redacted]

Username cornelluniversity[redacted]

Authentication Credentials

Configure credentials for the authentication method this user will use.

Token [redacted]

URL

	URL
General Pattern	https://ALB-CNAME.boomi.xxxx.cucloud.net/ws/rest/aws-molecule-heartbeat/heartbeat
Production	https://prod-asg.boomi.xxxx.cucloud.net/ws/rest/aws-molecule-heartbeat/heartbeat

```
curl --location \
         --request GET 'https://prod-asg.boomi.xxxx.cucloud.net/ws/rest/aws-molecule-heartbeat/heartbeat' \
     --header 'Authorization: Basic TOKEN_IN_BASE64'
```